REMARKS

The abstract has been amended, the specification has been amended, the title has been amended and the underlining of the section headings has been removed, as requested. The underlined section headings in this amendment merely reflect the newly inserted text (minus the underlining) in accordance with Patent Office rules for filing amendments.

Claims 6, 7 and 15 have been amended to address the rejection under 35 USC § 112, second paragraph. It appears that claim 30 was open to the same objection as claim 6 and claim 30 has therefore also been amended. Withdrawal of this rejection is respectfully requested.

Claim 1 has been amended to include the features of the original claim 14. Claim 14 depended on claim 1 via claim 5, and certain features of claim 5 have also been incorporated in amended claim 1. Amended claim 1 also recites the feature, not present in the original claim 14, that the next-section-locating information is stored within the processor.

In paragraphs 29 and 30 of the Office Action, the Examiner rejected independent claims 1 (now amended) and 37 (also amended) as being anticipated by Faraboschi et al. (US 5,870,576). Applicants traverse this rejection because Faraboschi et al. do not disclose (or suggest) that the next-section-locating information is stored within the processor, as in amended claims 1 and 37.

In Faraboschi Fig. 2, a main memory 110 corresponds to the "program memory" of amended claim 1. This main memory 110 is divided into two segments, namely a code pointer segment 130 and a code heap segment 140. The code heap segment 140 stores the compressed-form instructions. The code pointer segment 130 stores a "code pointer" 132 (32 bits in total) for each compressed-form instruction word. The first nine bits of the code pointer are a "mask" 150, which is equivalent to the "decompression key" in the present invention. The remaining 23 bits are a pointer "ptr" 152 representing an offset between the address at which the code pointer is stored in the main memory and the address at which the first compressed-form instruction of the instruction word is stored in the main memory. Thus, for example, in the case of the code pointer 132 stored at address 12345100, the first compressed-form instruction W00 is stored at address 14000300, and what is stored in the ptr 152 at address 12345100 is therefore the offset (+12ec80) between the addresses 14000300 and 12345100. The program counter 200 in Faraboschi points to successive locations in the code pointer segment 130 of the main memory 110. The program counter value (PC) is supplied to a cache refill state machine 204. If a cache miss occurs (i.e. the instruction word pointed to by the code pointer at the address specified by PC is not already present in decompressed form in the instruction cache 100 of the processor), the cache refill state machine accesses the code pointer at the address pointed to PC, and uses the offset 152 to locate the first compressed-form instruction of the instruction word concerned. The compressed-form instructions of the instruction word are then decompressed and loaded into the instruction cache 100.

The present invention is concerned with a processor that is capable of performing "on-the-fly" decompression. When a cache miss occurs, the decompression of the compressed-form instructions and the loading of the decompressed instructions into the instruction cache are time-critical operations for the processor. In the present invention, these time-critical operations are facilitated by holding next-section locating information, used to locate the position in the program memory of a next compressed section following the compressed section corresponding to the most-recently-accessed cache block, within the processor itself. In Faraboschi, the next-section locating information is the ptr 152, which is held outside the processor in the main memory 110. As a result, there is a delay in accessing this information and consequently a delay in commencing the decompression and loading processes.

The decompression processes of the present invention and Faraboschi are fundamentally different. In Faraboschi, the aim is to have the program counter incremented by a fixed amount each instruction cycle, so that it simply moves through sequential locations in the code pointer segment 130 of the main memory 110. This provides the advantage that control of the program counter is simple and quick. However, the penalty for this is that there must be an address mapping operation from the location in the code pointer segment 130 to the relevant location in the code heap segment 140, and this causes some delay.

In the present invention, on the other hand, the program counter points directly to the position of the next set of compressed-form instructions in the program memory. This means that the program counter cannot simply be incremented by a fixed amount each instruction cycle. Thus, there is a penalty in terms of complexity of program counter manipulation. However, the advantage of doing this is that the program counter is maintained consistently at the position in the program memory where the next instruction is stored in compressed form, as specified by claim 1. Accordingly, this instruction can be accessed with minimum delay.

In view of the fundamentally different approaches in Faraboschi and the present invention, a skilled worker would not seek to modify Faraboschi to arrive at a processor in accordance with amended claim 1, since to do so would require him to discard the entire approach taught by Faraboschi. There is no motivation in Faraboschi for him to do so. Accordingly, the amended claim 1 is neither anticipated by, nor obvious in light of, Faraboschi.

Claim 37 has been amended to include the same limitations as the amended claim 1, and accordingly this claim is also believed to be allowable over Faraboschi. For these reasons, withdrawal of the rejection of claims 1 and 37, and the related rejected dependent claims is respectfully requested.

New claim 38 and dependent claims 39-42 are also directed to a processor. New dependent claims 39-42 correspond respectively to original claims 5, 6, 9 and 10. Claim 38 is based on a combination of the original claims 1, 2 and 4. This claim recites that imaginary address information is stored in the program memory along with the compressed-form instructions. The processor uses the imaginary address information to assign imaginary addresses to the instructions when they are decompressed and loaded into the instruction cache. The cache loading unit determines which cache block of the instruction cache to store the decompressed instructions in, based on the imaginary addresses.

These features of new claim 38 are not disclosed or suggested in Faraboschi. In Faraboschi, the main memory has a code pointer segment 130 in which the offsets 152 mentioned above are stored. However, these offsets are not equivalent to the imaginary address information of new claim 38. The location in the instruction cache 100 at which the decompressed instructions of each instruction word will be stored is determined directly by the program counter PC in Faraboschi. The location is not determined by any information stored in the program memory (main memory 110), as in claim 38.

There are certain advantages in storing the imaginary address information in the program memory along with the compressed-form instructions. In particular, it is possible to make more efficient utilization of the cache, for example, to minimize the number of cache misses, as described in the specification at page 34, lines 6 to 13. For these reasons, claims 38-42 are allowable.

Claims 27 and 36 include minor clarifying amendments. First, although the compressed program produced by the compression method must be present in a program memory of the processor at execution time, the compression method need not itself involve the step of storing the compressed program in that program memory. That storage step could be carried out separately, for example, by an end-user who receives the compressed program on a disk or CD-rom. Accordingly, amended claims 27 and 36 refer to "outputting a compressed program storable in the program memory and comprising the compressed-form instructions together with imaginary address information...". Secondly, as indicated in the specification at page 32, lines 5 to 17, the imaginary address information may specify the assigned imaginary address of only one of the original instructions, and this is now clarified by the amended claims 27 and 36.

The Examiner rejected independent claims 27 and 36 as originally filed as being obvious in light of Faraboschi (paragraphs 85 to 90 and 135 to 143 of the Office Action). Applicants traverse. As noted above, the offsets 152 stored in the main memory 110 (program memory) in Faraboschi are not equivalent to the imaginary address information of claims 27 and 36. It would not be obvious to modify the offsets in Faraboschi to arrive at imaginary address information in accordance with the present invention, since as noted above the approaches adopted in Faraboschi and the present invention are fundamentally different. Withdrawal of the rejection of claims 27 and 36, and their related rejected dependent claims, is respectfully requested.

New claim 43 is a further independent claim directed to a computer-readable recording medium storing a compressed program embodying the present invention. The wording of this claim resembles that of claims 27 and 36. As noted above, the compressed program produced by the method of claim 27 need not be stored in the first place in the program memory of the processor, and claim 43 is intended to cover any recording medium in which the compressed program may be stored prior to transfer into the program memory. New claims 44 to 51 correspond respectively to claims 28 to 36.

For the foregoing reasons, applicants believe that this case is in condition for allowance, which is respectfully requested. The examiner should call applicants' attorney if an interview would expedite prosecution.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By
Patrick G. Burns
Registration No. 29,367

June 25, 2004

300 South Wacker Drive
Suite 2500
Chicago, Illinois 60606
Telephone: 312.360.0080
Facsimile: 312.360.9315
P:\DOCS\0808\65559\584403.DOC